I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on April 11, 2005.

Marilyn R. Khorsandi
Marilyn R. Khorsandi

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	Bilibin, Paul et al.
Application No.	:	09/684,861
Filed	:	October 6, 2000
Title	:	Apparatus, Systems and Methods for Determining Delivery Time Schedules for Each of Multiple Carriers
Grp./Div.	:	3623
Examiner	:	Van Doren, Beth
Docket No.	:	PSTM0024/MRK

DECLARATION UNDER 37 C.F.R. 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

We, Paul Bilibin and Jinyue Liu, do hereby declare and state as follows:

1. We, Paul Bilibin and Jinyue Liu, are co-inventors of the invention described and claimed in the above-referenced patent application. We each have reviewed the Office Action mailed January 11, 2005 in the above-referenced application. We have also reviewed U.S. Patent Number 6,064,981, Barni et al. ("*Barni*"), which was relied upon by the Examiner in rejecting Claims 1-12 and 14-18 under 35 U.S.C. §102(e) and in rejecting Claim 13 under 35 U.S.C. §103(a). This Declaration is filed under 37 C.F.R. §1.131 to substantiate our invention of the subject matter claimed in the present application prior to June 17, 1999, the reported filing date of the *Barni* reference.

2. Prior to June 17, 1999, we invented a multi-service, multi-carrier, Internet-enabled server-based shipping system for use by small volume shippers such as small businesses and home offices. The concept behind this multi-carrier, multi-service, Internet-based shipping system was to provide shipping users ("shippers") with a cross-comparison of shipment rating, service options, delivery schedules and other services by each of the multiple carriers for each of multiple services so that a shipper could compare multiple services offered by the multiple carriers and select one service offered by one of the multiple carriers to ship a parcel. When this shipping system was first conceived, we worked for MoveIt! Software Inc.

("MoveIt!"), a company founded in 1997. Later, MoveIt! became iShip.com Inc., which eventually merged with Stamps.com Inc., and which is currently a wholly owned subsidiary of United Parcel Service ("UPS"). As of the latest date of this Declaration, iShip Inc. and Stamps.com Inc. are joint owners in common of the subject invention. Through all of the aforementioned corporate changes, we continued to develop and test the above-mentioned multi-carrier Internet-based shipping system that ultimately led to the filing of the above-referenced patent application.

3. As claimed in one form or another in independent Claims 1, 4 and 7 of the present application, as amended, the shipping management system, and the methods and computer program products for that system, that we invented prior to June 17, 1999, provided a shipping management computer for determining a respective potential cross-comparison delivery schedule, wherein said respective cross-comparison delivery schedule comprises a plurality of respective service-specific, carrier-specific delivery schedules to ship the particular respective parcel from a first address to a second address, and wherein each respective service-specific, carrier-specific delivery schedule corresponds to a respective particular delivery service of a plurality of delivery services offered by a respective particular carrier of a plurality of carriers. In support of the foregoing statement, we hereby submit, attached hereto as **Exhibit A**, a true and correct copy of pages 1, and 14-19 of a file copy of a Project Roxbury Implementation Requirements document; The Project Roxbury Implementation Requirements document was kept confidential within MoveIt!. The Project Roxbury Implementation Requirements document was prepared pursuant to an agreement with College Enterprises Inc. ("CEI") whereby MoveIt! was to install, operate, support, debug and nurture a Beta test version of an early prototype of the above-identified shipping system at a selected college campus. The purpose of the Beta test was to experiment with the early stage prototype system to determine if it worked, whether it would work over the Internet, and to identify and resolve problems and issues with the system. The CEI Beta Test Agreement provided for a 50/50 revenue sharing business model of gross profits. A true and correct copy of the CEI Beta Test Agreement with MoveIt! is attached hereto as **Exhibit B**. The finalization date of the Project Wolverine Design Document was prior to June 17, 1999.

4. As claimed in one form or another in independent Claims 1, 4 and 7 of the present application, the multi-carrier shipping management system, and the methods and computer program products for that system, that we invented prior to June 17, 1999, provided remote access by multiple shipping users via the Internet. In support of the foregoing

statement, we hereby submit, attached hereto as **Exhibit C**, a true and correct copy of pages 33-34 of the Project Wolverine Design Document. The Project Wolverine Design Document was kept confidential within MoveIt!. The Project Wolverine Design Document was prepared pursuant to an agreement with College Enterprises Inc. ("CEI") whereby MoveIt! was to install, operate, support, debug and nurture a Beta test version of an early prototype of the above-identified shipping system at a selected college campus. The purpose of the Beta test was to experiment with the early stage prototype system to determine if it worked, whether it would work over the Internet, and to identify and resolve problems and issues with the system. As previously mentioned above, the CEI Beta Test Agreement (a true and correct copy of which is attached hereto as **Exhibit B**) provided for a 50/50 revenue sharing business model of gross profits. The finalization date of the Project Wolverine Design Document was prior to June 17, 1999.

5. As claimed in dependent Claims 2, 5, 8, and 15 of the present application, as amended, the remotely accessible, multi-carrier, Internet-enabled shipping management system, and the methods and computer program products for that system, that we invented prior to June 17, 1999, provided for calculation of shipping rates for a plurality of services offered by a plurality of carriers. In support of the foregoing statement, we hereby submit, attached hereto as **Exhibit D**, a true and correct file copy of Pages 140-146 of the Project Wolverine Design Document.

6. As claimed in independent Claim 10 of the present application, the remotely accessible, multi-carrier, Internet-enabled shipping management system, and the methods and computer program products for that system, that we invented prior to June 17, 1999, provided for allowing a user to request a package delivery service by providing shipping specifications; receiving said shipping specifications from said user; identifying, from a plurality of carriers, a subset of carriers based on said shipping specifications, each of said subset of carriers being capable of satisfying said shipping specifications by providing said package delivery service to said user; identifying a first carrier from said subset of carriers and a first set of shipment types provided by said first carrier; determining a first set of delivery schedules according to which said first carrier would be able to satisfy said shipping specifications, each one of said first set of delivery schedules corresponding to at least one of said first set of shipment types and comprising a delivery date and a delivery time; calculating a first set of service charges by said first carrier, each one of said first set of service charges calculated based upon at least one of said first set of shipment types provided by said first carrier; displaying to the user said first set

of delivery schedules, said first set of service charges, and said first set of shipment types; identifying a second carrier from said subset of carriers and a second set of shipment types provided by said second carrier; determining a second set of delivery schedules that said second carrier is capable of providing to said user, each one of said second set of delivery schedules corresponding to at least one of said second set of shipment types and comprising a delivery date and a delivery time; calculating a second set of service charges by said second carrier, each one of said second set of service charges calculated based upon at least one of said second set of shipment types provided by said second carrier; and displaying to the user said second set of delivery schedules, said second set of service charges, and said second set of shipment types. In support of the foregoing statement, we hereby reference **Exhibits A and D**, attached hereto, as previously described above.

7. From the time of our invention until the filing date on October 6, 1999 of a first provisional application on which priority for the present application is based in part, and thereafter, we continued our effort to refine the operation of the system and resolve problems with the system. In support of the foregoing statement, we hereby submit, attached hereto as **Exhibit E**, a true and correct copy of a stream of emails exchanged between various members of the MoveItl/iShip development team.

8. We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that the statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001, Title 18 of United States Code and that such willful false statements may jeopardize the validity of the application or any corresponding U.S. patent.

Date: April 2005

Paul Billbin

Date: April 7, 2005

Jinyue Liu

iShip.com 1.1

Implementation Requirements

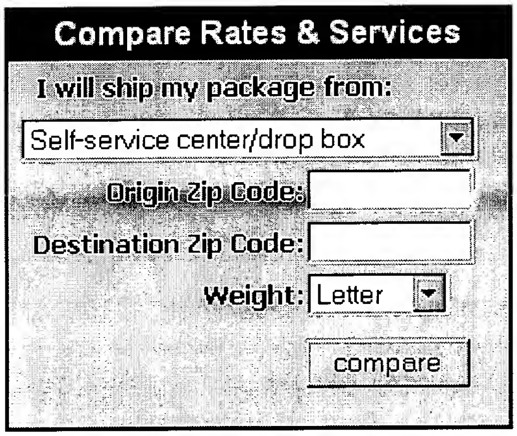
October 17, 1998

iShip.com Confidential

iShip.com, Inc.
2515 140th Ave. NE - Suite E110
Bellevue, Washington 98005
Phone: (206) 602-5028
Fax: (206) 602-5021

4.2.2. COMPARITOR

1. The Comparitor controls consists of the following:
    1.1. Ship Location Type drop down list.
        1.1.1. This will list all available Ship Location types. See Location and Package Information page for more information regarding Ship Location classes.
    1.2. Origin Zip Code Field
    1.3. Destination Zip Code Field.
    1.4. Weight drop down list
        1.4.1. The list will contain:
            1.4.1.1. A "Letter" selection
                1.4.1.1.1. If "Letter" is selected the Packaging Type will be set to Carrier Letter.
                1.4.1.1.2. If "Letter" is selected the weight will be set to 0.5 lbs.
            1.4.1.2. Weights from 1 to 150 lbs.
                1.4.1.2.1. If a specific weight is selected the Packaging Type will be set to Carrier Box.
    1.5. "Compare" button


Compare Rates & Services

I will ship my package from:

Self-service center/drop box

Origin Zip Code:

Destination Zip Code:

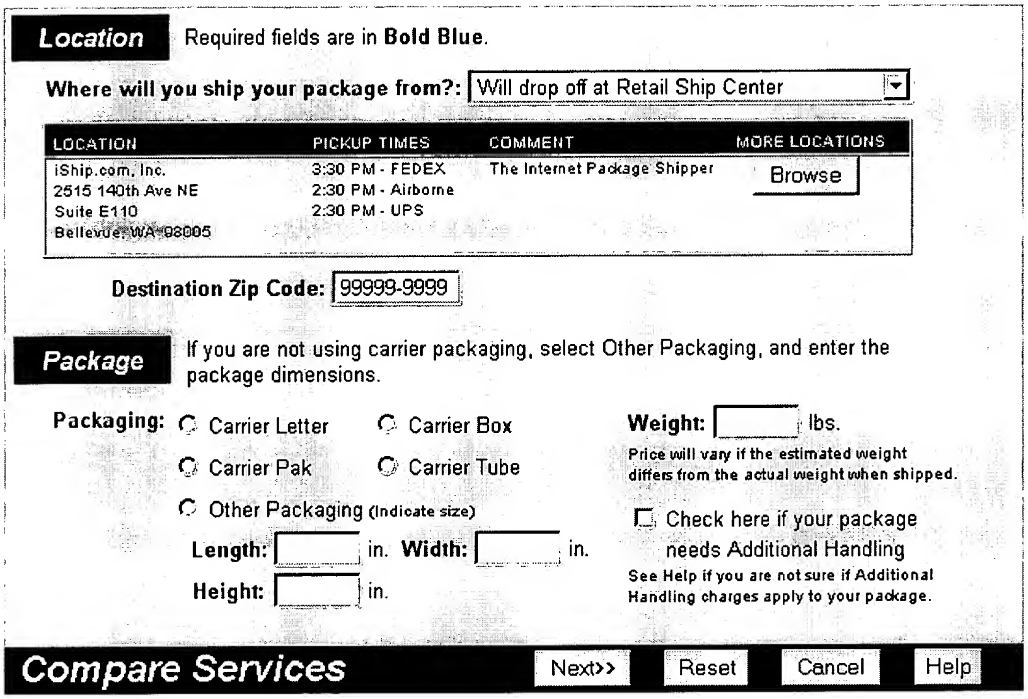
Weight: Letter

Illustration 5: Comparitor

2. If the Ship Location type is of a "customer drop off" class, when the "Compare" button is pressed the user will be sent to the Delivery Times and Rates page using the entered values to determine rates.
3. If the Ship Location type is of a "customer drop off" class, when the "Compare" button is pressed:
    3.1. The Drop Off Locator will be opened in a pop-up window.
        3.1.1. The Origin Zip Code and Ship Location type values will be used as parameters for the Drop Off Locator, such that the Drop Off Locator will be displayed with a list possible choices when it is first displayed.
    3.2. After a selection is made in the Drop Off Locator, the user will be sent to the Delivery Times and Rates page using the entered values to determine rates.

4.2.3. LOCATION AND PACKAGE INFORMATION PAGE

1. The first page of Compare Services will contain the Location and Package Information areas.
2. In the Location area of the page the first control is the Ship Location Type drop down list.
   2.1. While there are 4-5 selections in the list, there are only two classes of locations.
      2.1.1. Those which have a database of specific locations, from which a specific location from the available locations must be selected to determine rates, such as an "iShip Center" or "MBE Ship Center", collectively to be called "ship centers".
      2.1.2. The second class of location are those which may or may not be tied to a database of specific locations, and from which a specific location need not be selected to determine rates, such as a "drop box", "carrier counter" or "call for pickup", collectively to be called "customer drop offs".
   2.2. If a "ship center" is selected as the shipping location, the Location area will display three elements:
      2.2.1. A table with the Location Address, Pickup Times and Comments Area.
         2.2.1.1. If no location is selected the table will display a message asking the user to select a location.
      2.2.2. A Browse button which will display the Drop Off Locator in a Pop-up window.
      2.2.3. A destination Zip Code field.



Location Required fields are in **Bold Blue**.

Where will you ship your package from?: Will drop off at Retail Ship Center ▾

LOCATION	PICKUP TIMES	COMMENT	MORE LOCATIONS
iShip.com, Inc. 2515 140th Ave NE Suite E110 Bellevue, WA 98005	3:30 PM - FEDEX 2:30 PM - Airborne 2:30 PM - UPS	The Internet Package Shipper	Browse

Destination Zip Code: 99999-9999

Package If you are not using carrier packaging, select Other Packaging, and enter the package dimensions.

Packaging: ○ Carrier Letter ○ Carrier Box ○ Carrier Pak ○ Carrier Tube ○ Other Packaging (Indicate size)

Length: ____ in. **Width:** ____ in.

Height: ____ in.

Weight: ____ lbs.

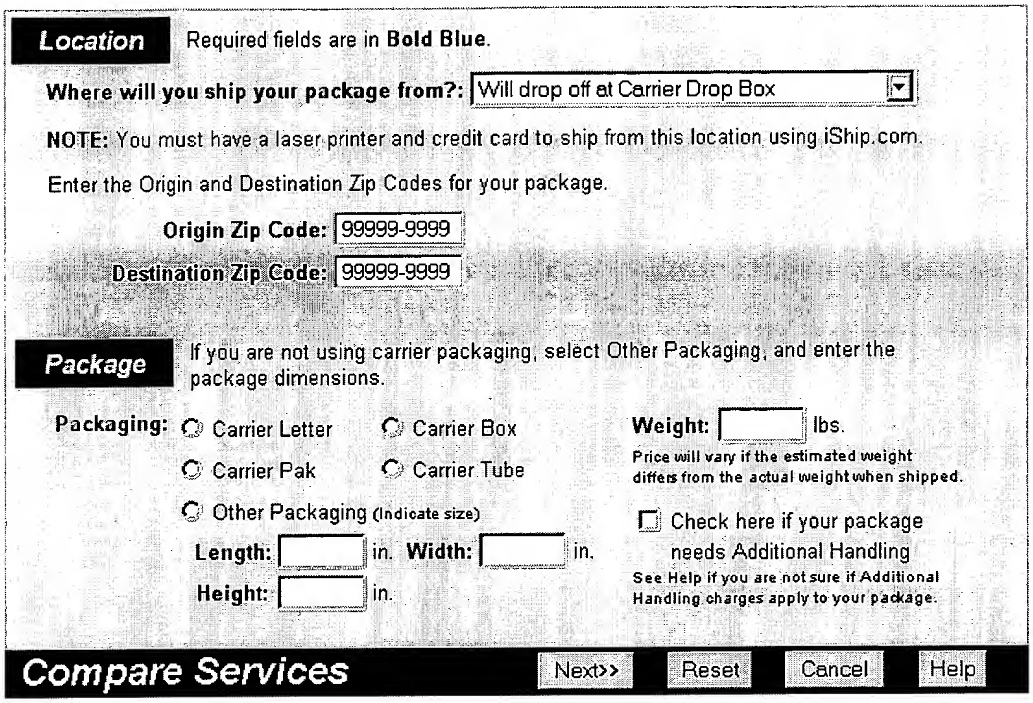
Price will vary if the estimated weight differs from the actual weight when shipped.

□ Check here if your package needs Additional Handling

See Help if you are not sure if Additional Handling charges apply to your package.

Illustration 6: Location and Package page - controls for "ship center"

   2.3. If a "customer drop off" is selected as the shipping location The Location Area will display two elements.
      2.3.1. Origin Zip Code Field
      2.3.2. Destination Zip Code Field.
   2.4. For either class of shipping location, if an iShip Shipping Station will not be present at the location, a notice will be displayed to the customer telling them that they must have a laser printer to ship using this location.
   2.5. If the user is Logged On the Location area will default to the users Preference.
   2.6. If the Preference is a "customer drop off" location, the Origin Zip Code will be populated with their default Zip Code.



Required fields are in **Bold Blue**.

Where will you ship your package from?: Will drop off at Carrier Drop Box

NOTE: You must have a laser printer and credit card to ship from this location using iShip.com.

Enter the Origin and Destination Zip Codes for your package.

Origin Zip Code: 99999-9999

Destination Zip Code: 99999-9999

If you are not using carrier packaging, select Other Packaging, and enter the package dimensions.

Packaging: ○ Carrier Letter ○ Carrier Box ○ Carrier Pak ○ Carrier Tube ○ Other Packaging (Indicate size)

Length: in. **Width:** in. **Height:** in.

Weight: lbs.

Price will vary if the estimated weight differs from the actual weight when shipped.

□ Check here if your package needs Additional Handling

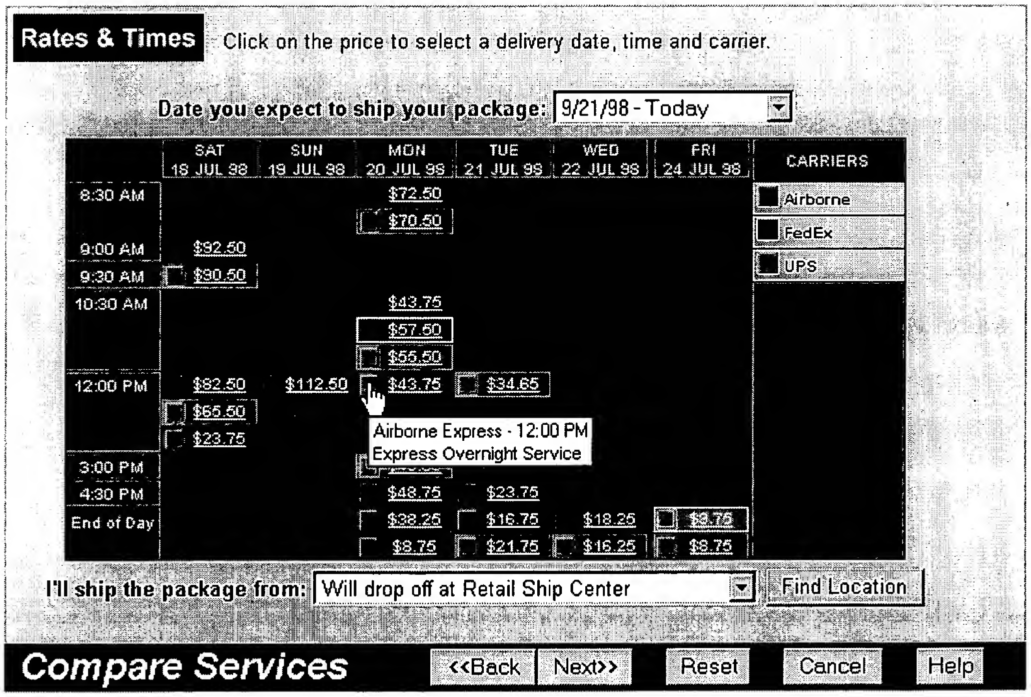
See Help if you are not sure if Additional Handling charges apply to your package.

Illustration 7: Location and Package page - controls for "customer drop off"

3. In the Package area of the page there are the following controls or control groups:
    3.1. Packaging
        3.1.1. This includes Length, Width and Height which are required for Other Packaging. For all carrier packaging the field will be auto filled with "letter", "pak", etc.
    3.2. Weight
    3.3. Additional Handling
4. The Next button will go to the Delivery Times and Rates page.
5. Reset will clear the fields and remain on the Location and Package page.

4.2.4. DELIVERY TIMES AND RATES PAGE

1. The Delivery Times and Rates page will contain the following controls, control groups and display items.
   - 1.1. The Expected Ship Date drop down will contain seven entries from the current date to Plus six days.
     - 1.1.1. The format is "M/D/YY - Day name", w/ "Today" and "Tomorrow" substituted appropriately.
     - 1.1.2. When a new date is selected the page will refresh with new rates and times from the Server.
   - 1.2. A Ship Location Type drop down list will be located below the rate grid.
     - 1.2.1. The list of locations will be the same as the Ship Location Type drop down list in the Location area of the Package and Location page.
       - 1.2.1.1. If the Shipping Location class is a "ship center", a "Find Location" button will be displayed next to the drop down.
       - 1.2.1.2. The "Find Location" button will open the Drop Off Locator in a pop-up window.
     - 1.2.2. When a selection from the list is selected the rate grid will update to reflect any rate changes or surcharges.
   - 1.3. The Rate Grid display will display the following:
     - 1.3.1. Valid delivery dates, across the top of grid, for the selected Ship Date.
     - 1.3.2. Sorted, valid delivery times for all valid dates down the left side of the grid.
     - 1.3.3. Color coded by carrier rates cells for each available rate, by date and time.
       - 1.3.3.1. Each cell will contain an small image which will contain ALT text. The ALT text shall contain the full carrier name and the full carrier service name.
2. The Back button will return the user to the Location and Package Page.
3. The Next button will go to the Service Option page.
   - 3.1. A Rate must be selected before the user can go to the Service Option Page.
4. If a user returns to this page from the Service Option Page:
   - 4.1. Any selected Service Options will effect the displayed rates.
   - 4.2. Any selected Service Options will be displayed as abbreviations below the Shipping Location radio buttons.
5. Reset will clear all fields and return to the Location and Package page.



Click on the price to select a delivery date, time and carrier.

Date you expect to ship your package: 9/21/98 - Today

	SAT 18 JUL 98	SUN 19 JUL 98	MON 20 JUL 98	TUE 21 JUL 98	WED 22 JUL 98	FRI 24 JUL 98
8:30 AM			$72.50			
			$70.50			
9:00 AM	$92.50					
9:30 AM	$90.50					
10:30 AM			$43.75			
			$57.50			
			$55.50			
12:00 PM	$82.50	$112.50	$43.75	$34.65		
	$65.50					
	$23.75					
3:00 PM			[illegible]			
4:30 PM			$48.75	$23.75		
End of Day			$38.25	$16.75	$18.25	$9.75
			$8.75	$21.75	$16.25	$8.75

CARRIERS
- ■ Airborne
- ■ FedEx
- ■ UPS

Airborne Express - 12:00 PM
Express Overnight Service

I'll ship the package from: Will drop off at Retail Ship Center

Illustration 8: Delivery Times and Rates page

4.2.5. SERVICE OPTIONS PAGE

1. The Service Options page will contain the Service Options and the Single Day Rate Grid.

2. The Service Options supported are:

2.1. Loss Protection (Declared Value)

2.1.1. If the user selects Declared Value they must enter a value of greater than $100.00, and equal to or less than $50,000.00.

2.1.2. The page will update the Single Day Rate Grid with changes in individual rates.

2.2. E-Mail Delivery Notification

2.2.1. This is two controls - a checkbox and an "E-Mail Others..." button.

2.2.2. If the checkbox is checked the rates will be updated to reflect the charges.

2.2.3. If the button is pressed the following will occur:

2.2.3.1. If the checkbox was not checked, it will be checked.

2.2.3.2. The "E-Mail Others..." page will be displayed in a pop-up window. See below for description of the E-Mail Others page.

2.3. Verbal Confirmation

2.3.1. If the checkbox is checked the rates will be updated to reflect the charges.

2.3.2. If checked the application will use the Return Address Phone and Name as the values required by UPS, if the user elects to ship the package.

2.4. "Service must be guaranteed"

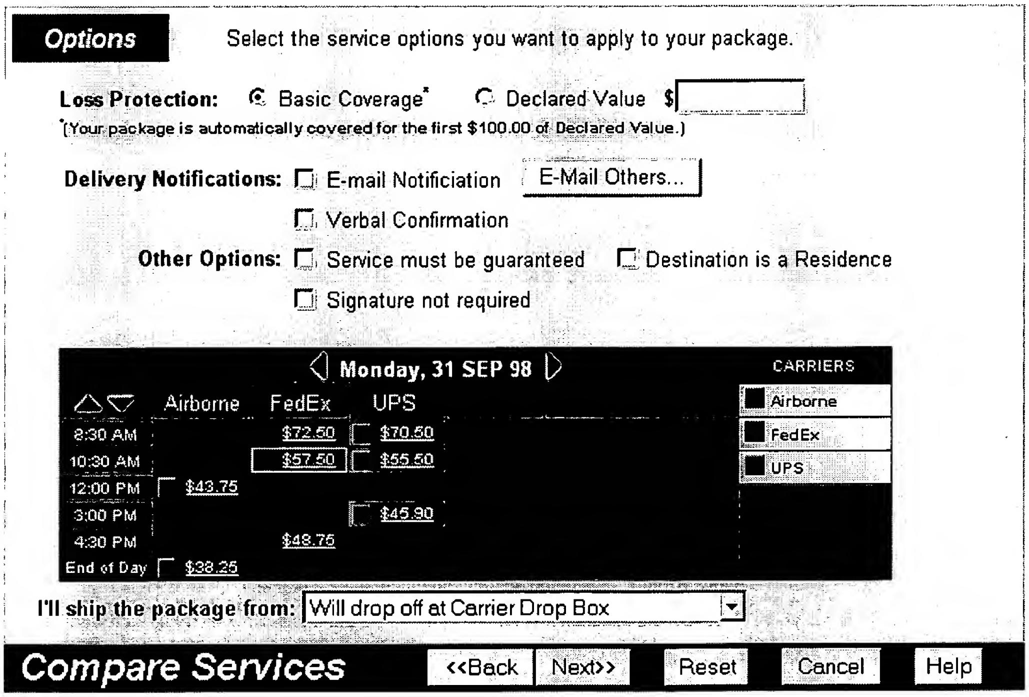
2.4.1. If the checkbox is checked the rates will be updated to remove any service which is not guaranteed.

2.5. "Destination is a Residence"

2.5.1. If the checkbox is checked the rates will be updated to reflect the charges and remove any services which are no longer valid.

2.6. "Signature not Required"

2.6.1. If the checkbox is checked no change will be applied to the rate grid. This is a FedEx only flag and does not effect any other carrier or any carrier rate.



Options Select the service options you want to apply to your package.

Loss Protection: ● Basic Coverage* ○ Declared Value $

*(Your package is automatically covered for the first $100.00 of Declared Value.)

Delivery Notifications: □ E-mail Notificiation [E-Mail Others...]

□ Verbal Confirmation

Other Options: □ Service must be guaranteed □ Destination is a Residence

□ Signature not required

◁ Monday, 31 SEP 98 ▷

△▽	Airborne	FedEx	UPS
8:30 AM		$72.50	□ $70.50
10:30 AM		$57.50	□ $55.50
12:00 PM	□ $43.75		
3:00 PM			□ $45.90
4:30 PM		$48.75	
End of Day	□ $38.25		

CARRIERS

■ Airborne

■ FedEx

■ UPS

I'll ship the package from: Will drop off at Carrier Drop Box

Illustration 9: Service Options page

3. The Single Day Rate Grid contains the following elements:

3.1. The selected delivery date at the top of grid.

3.1.1. This is bordered by left and right arrow buttons. If a Button is pressed the date will either go back (left) one valid delivery date or move forward (right) one valid delivery date.

3.1.1.1. The range of valid delivery dates is determined by the Expected Ship Date.

3.2. Sorted, valid delivery times for all valid dates down the left side of the grid.

3.2.1. Above the delivery times are up and down arrow buttons. If a button is pressed the list of available times will scroll up or down appropriately, if and only if the list exceeds the grid display area.

3.3. A Ship Location Type drop down list will be located below the rate grid.

3.3.1. The list of locations will be the same as the Ship Location Type drop down list in the Location area of the Package and Location page.

3.3.1.1. If the Shipping Location class is a "ship center", a "Find Location" button will be displayed next to the drop down.

3.3.1.2. The "Find Location" button will open the Drop Off Locator in a pop-up window.

3.3.2. When a selection from the list is selected the rate grid will update to reflect any rate changes or surcharges.

3.4. Color coded by carrier rates cells for each available rate, by date and time.

3.4.1. Each cell will contain an small image which will contain ALT text. The ALT text shall contain the full carrier name and the full carrier service name.

4. The Back button will return the user to the Delivery Times and Rates page.

5. The Next button will go to the Summary page.

5.1. A Rate must be selected before the user can go to the Summary Page.

6. Reset will clear all fields and return to the Location and Package page.

2515 140th Avenue Northeast
Bellevue, WA 98005
Tel: 425.372.1512
Fax: 425.372.1502
stevet@moveit.net

MoveIt! Software, Inc.

On-line shipping for everyone, everywhere...

November 26, 1997

Donald B. Mask
President and Chief Executive Officer
College Enterprises, Inc.
21201 Victory Blvd., Suite 270
Canoga Park, CA 91303

RE: LETTER OF INTENT

Dear Don:

The following provides a summary of business terms for the installation and operation of *MoveIt!* Web-based shipping systems at CEI's Pulse Centers and Special Teams campuses.

Parties

College Enterprises Inc. ("CEI")
Attn: Donald B. Mask, President & CEO
21201 Victory Blvd., Suite 270
Canoga Park, CA 91303
Tel: 818-615-0560

MoveIt! Software, Inc. ("MoveIt!")
Attn: Stephen M. Teglovic, President & CEO
2515 140th Avenue Northeast
Bellevue, WA 98005
Tel: 425.372.1512

Business Proposition

CEI and *MoveIt!* will enter into a business alliance whereby *MoveIt!* will install Web-based shipping stations ("Stations") in Pulse Copy & Technology Centers ("Pulse Centers") and on Special Teams campuses. This process will be undertaken in three phases:

- **Phase One:** develop Stations, launch *MoveIt!* Web Client and Network Operations Center (NOC) for installation in Pulse Centers to serve students/low-volume shippers;
- **Phase Two:** integrate CEI's Special Teams debit cards to increase convenience of the payment process; and
- **Phase Three:** expand to include university departmental business.

Term

This business alliance will have an initial term of five years, subject to earlier termination by either party for breach by the other.

Exclusivity

Subject to the terms contained herein, CEI and *MoveIt!* will work exclusively with each other on offering shipping services on US college campuses listed in a schedule to be attached to the definitive agreement.

Revenue Sharing

CEI and *MoveIt!* agree to share 50/50 in the "gross profits" generated by package shipments that are the result of shippers using the *MoveIt!* Stations at Pulse Centers or elsewhere on colleges and universities represented by CEI. For purposes of this letter of intent, "gross profits" shall be defined simply as the difference between the

CEI customer charge (i.e. the shipper) and the actual shipping cost charged by the

third-party carrier. Each party shall be responsible for its own operating expenses relating to the Stations and *MoveIt!* shipping system. *MoveIt!* will not charge CEI or the Pulse Centers for the development costs of the technology or for the operation of the *MoveIt!* NOC.

Obligations of the Parties CEI will own/lease, operate and maintain the Stations at its expense. *MoveIt!* will provide technology, systems set-up and coordination, training and customer support for the Stations at its expense. CEI and *MoveIt!* will work together diligently to define the requirements, marketing plan and deployment plan for the three phases defined above.

Intellectual Property *MoveIt!* will own all intellectual property developed by it relating to the Stations and will provide CEI with a nonexclusive license (subject to the exclusivity provision above) to use such intellectual property in connection with the operation of the Stations in Pulse Centers.

Schedule *MoveIt!* agrees to install a Beta version (the "Beta") of the Station at a selected Pulse Center for the Phase One customer group approximately five to six months after execution of the definitive agreement. Said Beta will be in operation between 30 and 90 days based on acceptance criteria spelled out in the definitive agreement.

Conditions CEI agrees to deploy the Stations promptly in each new Pulse Center it opens during the term of the definitive agreement.

Letter of Intent The Parties understand and agree that these summary terms are non-binding and will not be effective until a definitive agreement is executed by the Parties and ratified by the Parties' respective Boards of Directors.

Don, we're truly excited about this business alliance and appreciate in advance your and your colleagues' help during the development process and beta testing. Would you kindly acknowledge your agreement with the terms herein by signing in the space provided below and I will instruct our attorney to begin drafting a definitive agreement. We look forward to a long and rewarding partnership with you and CEI.

Sincerely,

Stephen M. Teglovic

Stephen M. Teglovic
President & CEO

Acknowledged and agreed on December 6, 1997

By: ____________________
Donald B. Mask, President & CEO
College Enterprises, Inc.

SMT/ms

cc: *MoveIt! Software, Inc.* Board of Directors

6. Web Site Overview

Most web sites found on the Internet are collections of organized web pages which, although organized, can be traversed in an unordered fashion. These informational web sites can be discovered through multiple links and their pages bookmarked for later retrieval. The average web user has grown accustom to waiting for graphics and content to download over their limited internet connection, for all web sites are brought to the web user's browser through the same small pipe. The MoveIt! web site is different from the traditional web site, for it provides more than just an information-based collection of web pages. It provides a set of services and is compared more to an application that is running on the web user's machine. Applications have a whole different set of performance requirements in order to be perceived acceptable to the user. Because of this comparison to an application, we have had to mimic the flow that is associated with an application while working with the limitations imposed by a web browser's abilities and connections across the internet.

The web applications, which the MoveIt! web site provides, have had to overcome the unique set of challenges are imposed when designing a tool perceived as both an application and as a web site. Some of the solutions we implemented are:

- Keeping frequently used information in a web page cache and web browser cookies.
- Maintaining an application's state with client cached information.
- Providing secure information transactions with secure socket layers and session keys.
- Using pop-up browser windows to maintain an application's state while performing subtasks.
- Validating forms on the client side to stop simple errors from requiring a server round trip.

Because of the nature of a web site, the design needs to be both flexible and extendible. Some of the ideas we have incorporated into our design are:

- Stateless program flow to allow for interaction with multiple web servers.
- Code reuse with shared set of commonly used JavaScript functions and HTML.
- Data driven screen forms

6.1 Web Site Performance

Speed is the measure of performance for which the MoveIt! web site must optimize. Due to this requirement, the web site applications use a number of techniques to minimize the number of client web browser and web server transactions. One of the slowest processes of the web site is the transferring of information between web browser client and web server. Every time a web user clicks on a hypertext link, a round trip of information to the server must take place for the retrieval of content for the next web page. The following areas explain methods of improving the web site performance.

6.1.1 Client Form Validation

When a user puts information into a form on a web page, they will need to submit it back to the web server to have their data validated and processed. To alleviate some unnecessary round trips, simple validation can be performed on the client browser through JavaScript. The types of validation that can be performed on the client include:

- Verifying there is content in required information fields
- Checking for the "@" symbol in an email address.
- Preprocessing alpha or numeric only separation.
- Required number of password characters
- Changing password content confirmation

If these simple checks find invalid or missing data, they can immediately be brought to the attention of the web user rather than having their form make the round trip only to find that a simple mistake.

The process for performing the client validation involves JavaScript, which is called in one of several methods. The JavaScript can be called when a submit form button is clicked or from a number of different events like OnBlur, which triggers when the current field loses focus. An example of validation is the confirmation of a user's password. Passwords are entered into inputs of type "password", which have the added security measure of echoing typed characters as asterisks. Below is some sample HTML for a password input, a confirmation password input, and a submit button.

```
<FORM NAME="Test">
Password: <INPUT TYPE=PASSWORD NAME="Password1" SIZE=15> <BR>
Confirm: <INPUT TYPE=PASSWORD NAME="Password2" SIZE=15> <BR>
<INPUT TYPE=BUTTON VALUE="SUBMIT" OnClick="Confirm()">
</FORM>
```

The button input has an OnClick event that is set to call the function Confirm. The corresponding JavaScript verifies that the information matches and alerts the user if there is an inconsistency. Otherwise, the form is sent to the web server.

```
<SCRIPT LANGUAGE="JavaScript">
<!--
   function Confirm()
   {
      if( Test.Password1 != Test.Password2 )
      {
         alert( "Your password and confirmation do not match." );
      }
      else
      {
         Test.submit();
      }
   }
//-->
</SCRIPT>
```

Through JavaScript, a number of similar operations can be used to prevent server round trips on simple mistakes.

6.1.2 Client Information Caching

Once information has been validated and determined acceptable or it has been sent from the server once, it is desirable to not transfer the information every time it is required for display. Some data is stored in the client web browser, hidden from view. This "cached" data can be stored and retrieved through client JavaScript. For example, once a user has logged on to the MoveIt! system, their browser is passed a Session Key. The next transaction the web user makes will include their Session Key, which was held in their cache page. A new Session Key will be generated on the server and sent with the next page the client web browser receives. Another example of caching client information is the storing of data which is being accumulated over several pages and submitted at once for validation.

6.1.3 Selective Secure Transfers

When information needs to be sent between the client web browser and the web server and the information needs to be kept secure, the web page is through a secure socket. To make a socket secure, all messages sent through it are encrypted and then decrypted upon their arrival. Unfortunately, the process used to encrypt and decrypt information involves manipulating the data stream with complicated math functions and large numbers. This process, although very secure, is very slow. To alleviate the process for the web user, only selective information is transferred with a secure socket.

A large amount of the information which is contained within the web site is considered customer non-specific because most customers view it at some point. By keeping only customer specific information secure, the bulk of the data transfers are not penalized by security. One way this is accomplished is that frames that do not contain content that needs to be secure are not passed through a secure socket. Other broadly viewed objects in HTML web pages are graphics, which are referenced and transferred individually. Most graphics do not need to be kept secure and do not need to be sent through secure sockets. Because graphics are comprised of many more bytes than the

7. Rate Shop Application

7.1.1.1 Overview

The Rate Shop Application manages the on-screen-rate shopping. The user will be able to enter the following shipping information:

1- Origin Zip Code or Drop off Location

2- Destination Zip Code

3- Expected Drop off Date

4- Weight

5- Packaging type

6- Additional Handling

Once the above information is submitted, the MoveIt! Server components will validate the submitted shipping information; if the validation is successful, the Rating page will be displayed otherwise a descriptive error will be displayed.

7.1.1.2 Structure and Identification

Attribute	Value
Prefix	rate
Directory Name	Rateshop

Table 1. Rate Shop Structure and Identification

7.1.1.3 User Interface Design

The Rate Shop application entry page is the Rate Shop page.

The user enters here the necessary shipping information to get a rate for the package.

The user will be able:

1- To enter either an origin zip code or to select a location from a list of available drop off locations.

2- To enter destination zip code

3- To select expected drop off date from a list of the next seven working days

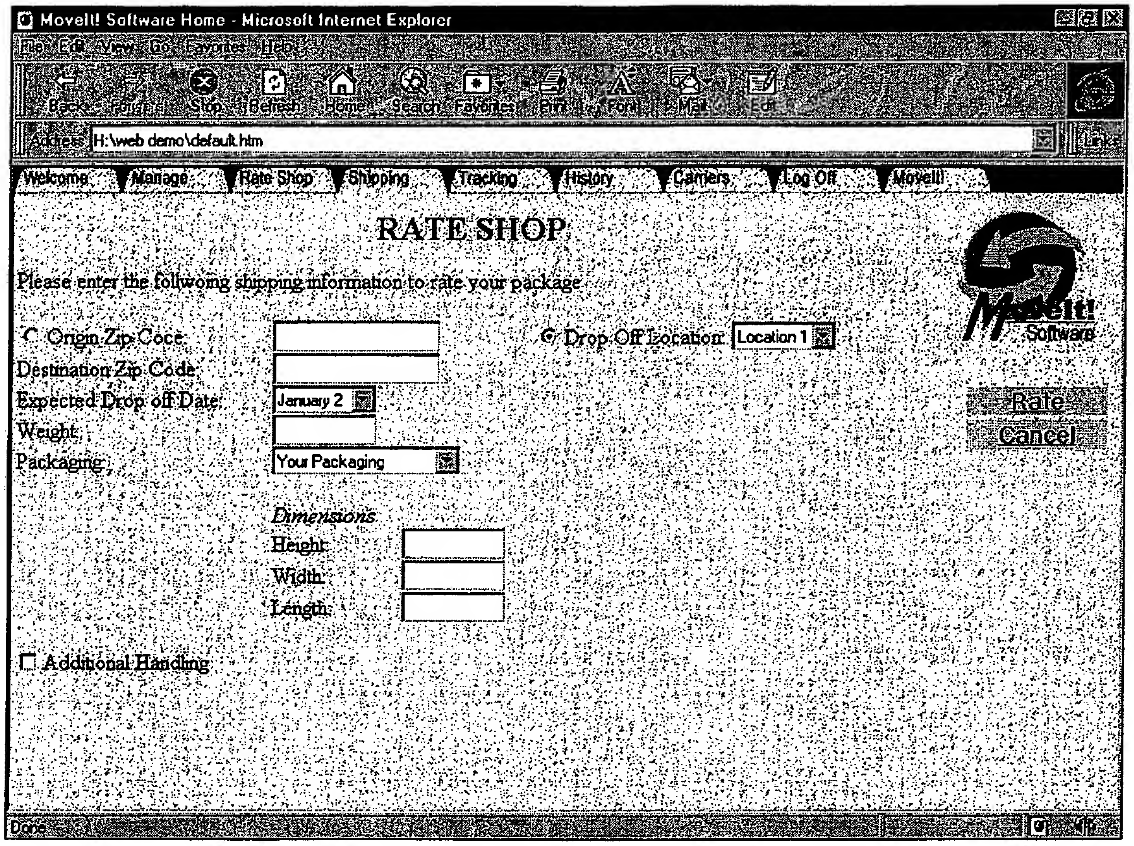
4- To select packaging type from a list of packaging types

If "Your Packing" selection is selected, then the user must enter:

- Height
- Width
- Length

5- To select Additional Handling

Figure 1 shows the Rate Shop page.



MoveIt! Software Home - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail Edit
Links
Welcome
Manage
Rate Shop
Shipping
Tracking
History
Carriers
Log Off
MoveIt!
MoveIt! Software
Rate
Cancel
Done

Address H:\web demo\default.htm

RATE SHOP

Please enter the follwong shipping information to rate your package

○ Origin Zip Coce ● Drop Off Location Location 1

Destination Zip Code

Expected Drop off Date January 2

Weight

Packaging Your Packaging

Dimensions

Height

Width

Length

□ Additional Handling

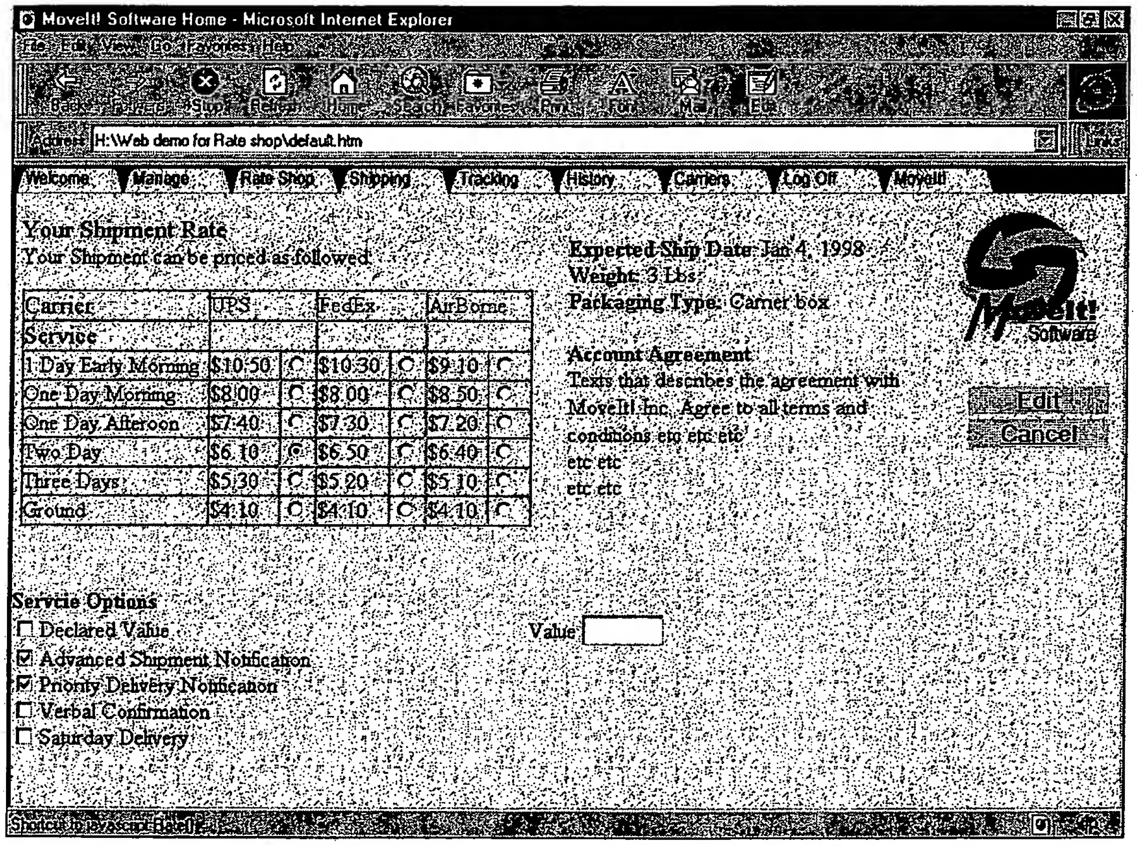
Figure 1. Rate Shop page

Once the information in the Rate Shop page are submitted and validated by MoveIt! Server components, the Rating page will be displayed.

The Rating page displays the following:

1. The expected ship date
2. Service options controls
    - Declared value
    - Outbound alert
    - Priority delivery notification
    - Verbal confirmation of delivery
    - Saturday delivery
3. Rate grid. It displays valid rates and services for submitted data and for each carrier
4. A condensed account agreement
5. Package weight

Figures 2. Shows a screen shot of the Rating page


MoveIt! Software Home - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail Edit
Links
Welcome
Manage
Rate Shop
Shipping
Tracking
History
Carriers
Log Off
MoveIt!
MoveIt! Software
Edit
Cancel
Shortcut to javascript:Rate0

Address H:\Web demo for Rate shop\default.htm

Your Shipment Rate

Your Shipment can be priced as followed

Carrier	UPS		FedEx		AirBorne	
Service						
1 Day Early Morning	$10.50	○	$10.30	○	$9.10	○
One Day Morning	$8.00	○	$8.00	○	$8.50	○
One Day Afteroon	$7.40	○	$7.30	○	$7.20	○
Two Day	$6.10	●	$6.50	○	$6.40	○
Three Days	$5.30	○	$5.20	○	$5.10	○
Ground	$4.10	○	$4.10	○	$4.10	○

Expected Ship Date: Jan 4, 1998
Weight: 3 Lbs
Packaging Type: Carrier box

Account Agreement
Texts that describes the agreement with MoveIt! Inc. Agree to all terms and conditions etc etc etc
etc etc
etc etc

Servcie Options
☐ Declared Value Value
☑ Advanced Shipment Notification
☑ Priority Delivery Notification
☐ Verbal Confirmation
☐ Saturday Delivery

Figure 2. Rating page

7.1.1.4 Data Flow Diagrams

The Data Flow diagram shows how the user accesses the Web Rate Shop Application, or the Rate Shop State "rate_Main", from the welcome screen by selecting the Rate Shop selection.

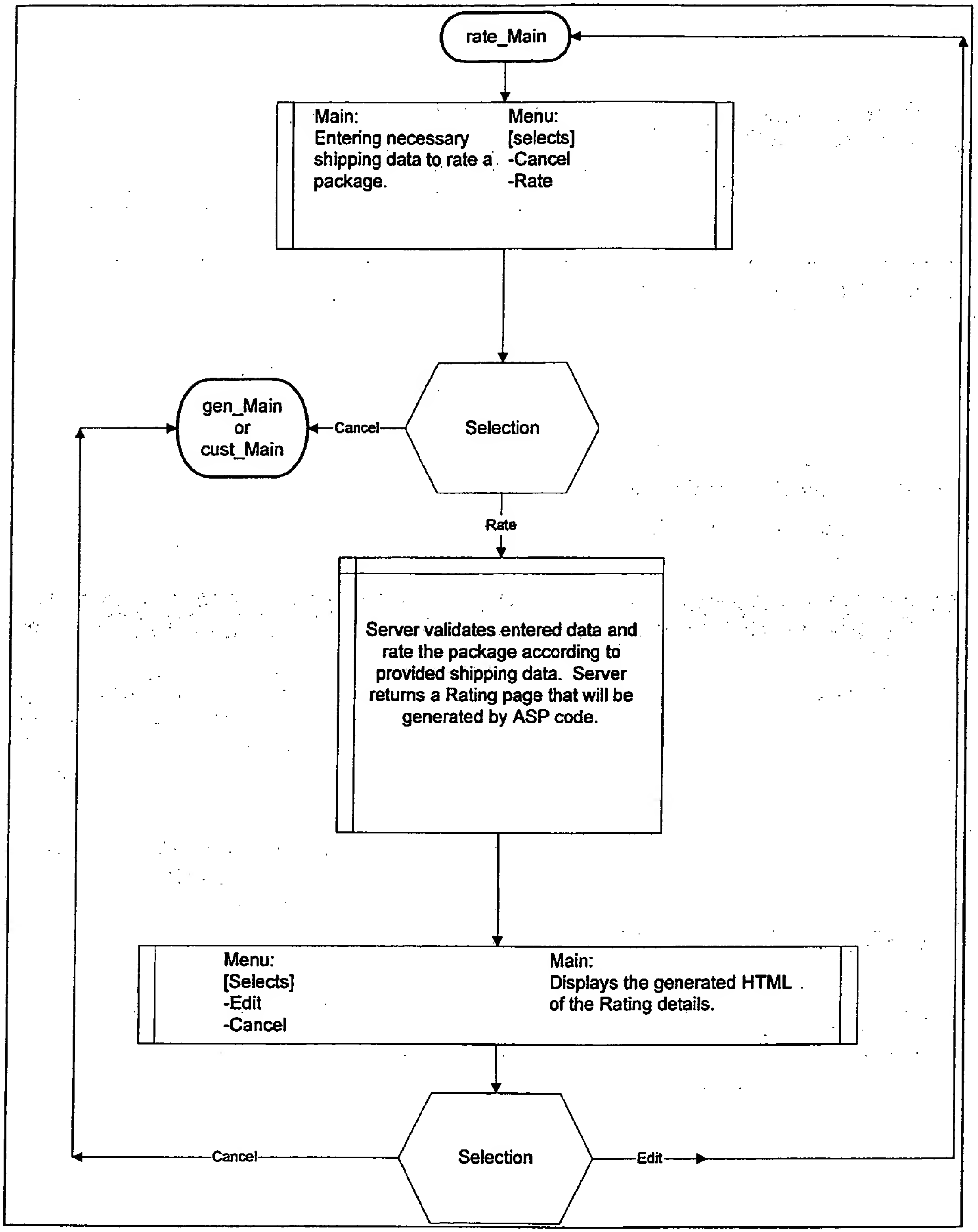

rate_Main
Main:
Entering necessary shipping data to rate a package.
Menu:
[selects]
-Cancel
-Rate
gen_Main
or
cust_Main
Cancel
Selection
Rate
Server validates entered data and rate the package according to provided shipping data. Server returns a Rating page that will be generated by ASP code.
Menu:
[Selects]
-Edit
-Cancel
Main:
Displays the generated HTML of the Rating details.
Cancel
Selection
Edit

Figure 3. Rate Shop data flow diagram

7.1.1.5 Transaction Sets

This table describes the transactions that the Web Reporting Application will have with the Web Server.

Transaction Sets	
Browser Request	Server Response(s)
Rate	Submit user's necessary shipping data to rate a package and return a HTML rating page

Table 2. Reporting Transaction Sets

7.1.1.6 External Dependencies

The physical dependencies that the Web Reporting Application has are:

Type	Name	Description
Utility Package Classes	NA	
COM Object	rate_IRating biz_IBusinessRules	Uses the carrier, service, drop-off site and destination information, weight, and dimensions to calculate a matrix of rate information per carrier and per service. The main function of this service will be to provide validation and calculation of small parcel packages.
Database Tables		Rate tables
Static Library	NA	

Table 3. Rate shop External Dependencies

From: Jinyue Liu [IMCEAEX-_O=VELLEB+2C+20INC+2E_OU=VELLEB_CN=RECIPIENTS_CN=JINYUE@iship.com]
Sent: Thursday, January 22, 1998 2:33 PM
To: MoveIt! Devel
Subject: Notes on optional tag of enum in IDL

Place the optional tag name of enum defined in an IDL in front of the definition {...} so
that this tag name will appear in the TLH file when you #import the TLB generated by this IDL.

for example,

```
typedef [v1_enum] enum EConnectType {dbs_kShipping, dbs_kTracking, dbs_kReport, dbs_kAccount } EConnectType;
```

when #import'ed, will generates

```
enum EConnectType

   dbs_kShipping = 0,
   dbs_kTracking = 1,
   dbs_kReport = 2,
   dbs_kAccount = 3
};
```

in the .TLH file.

But

```
typedef [v1_enum] enum {dbs_kShipping, dbs_kTracking, dbs_kReport, dbs_kAccount } EConnectType;
```

generates

```
enum __MIDL___MIDL_itf_Database_0000_0001
{
   dbs_kShipping = 0,
   dbs_kTracking = 1,
   dbs_kReport = 2,
   dbs_kAccount = 3
};
```

Following typedef without trailing tag will not compile.

```
typedef [v1_enum] enum EConnectType {dbs_kShipping, dbs_kTracking, dbs_kReport, dbs_kAccount } ;
```

-Jinyue

From: Paul McLaughlin [IMCEAEX-_O=VELLEB+2C+20INC+2E_OU=VELLEB_CN=RECIPIENTS_CN=PAULM@iship.com]
Sent: Tuesday, March 24, 1998 8:06 PM
To: MoveIt! Devel
Subject: Shipping Station Setup

Shipping Station Setup.doc

You now have two choices to when you want to use a shipping station:

1. Choose either the shipping station in the main conference room or the one in William's old office on the device rack on the bottom shelf on the left.

2. Put yourself through a little hell and setup your own PC as a shipping station and print labels on the shipping station in William's old office remotely. You don't need a scale.

You will have to follow this document to get option (2) to work:

<<Shipping Station Setup.doc>>

With either option, you should logon to NT using the following user information:
USER: sstation
Password: hamsandwich
Domain: moveit!

-LowRider

To setup a shipping station, do the following:

UPDATE NT:

- You must be running NT 4.0 with SP3.

UPDATE IE:

1. The Shipping Station requires IE 4.0.
2. Install IE 4.01 for NT. (This is on \\macarthur\scratch).
3. Close IE.
4. Logoff NT.
5. Logon NT as **"sstation"** with password **"hamsandwich"**.
6. After IE gets done updating your system, right-click IE on the desktop.
7. Select Properties.
8. Choose the Security dialog tab.
9. Select "Local intranet zone"
10. Click the "Low" radio button.
11. Choose the Advanced dialog tab
    Under the "Browsing" category
        Turn off, "Show welcome message each time I log on"
        Turn on, "Launch browser in full screen window"
12. Press OK to close the Properties dialog.

UPDATE CLIENT FILES:

1. Copy all the files from "N:\PAULM\SHIPPINGSTATION" to a local directory of your choice (using your Windows directory is an example).
2. "CD" into the directory you chose in the previous step, and type "SSSetup".

UPDATE MONITOR RESOLUTION:

- Set your screen resolution to 800x600, if you want the end-user viewing experience.

CONFIGURE PRINTERS:

- Setup a network printer driver to print to "`\\QA2\Eltron 2044`" by using the Add Printer wizard in Start/Settings/Printers. You may have to type in the net address of the printer to get the connection.

MISCELLANEOUS FINAL IE SETTINGS:

1. Launch IE.
2. The shipping station URL is "http://gambit/ShippingNetwork/ShippingStation". Either make this URL your IE startup page, or save a link to it in your Favorites. Make sure that the URL doesn't point to the Setup subdirectory of ShippingStation if you save it as your home page.
3. When IE launches, there may be an IE toolbar on the top. If so, right-click the toolbar and choose "Autohide".
4. If you chose 800x600 resolution, turn on "Autohide" in Start/Settings/Taskbar. IE should be truly "full screen" after doing so.

SETUP THE SHIPPING STATION APPLICATION

1. Tab to "Drop Off ID" and type "MoveIt! Headquarters".
2. Tab to "Password" and type "Battlezone" (yes, case sensitive).
3. Tab to "Shipping Station Name" and type whatever your heart desires.
4. Click the Submit button.
5. The Setup Control page doesn't work right now, so after it loads, press the "Goto Clerk Logon" button.
6. Logon as your email name for the Clerk ID (e.g. I logon as "paulm") and your password is initially "password".
7. Click Configuration
8. Click Printers
9. Choose "Eltron 2044" for the Label and Receipt printers.
10. Select the driver attached to "`\\qa2\eltron 2044`" for the "local driver" settings for the label and receipt printers.
11. You may leave the Report printer settings "not configured" for now as EOD isn't implemented yet.
12. Choose "Manual Override" from the list of Supported Scales. You may leave "Local Port" as is.
13. Click the "Submit New Configuration" button.
14. Click the Logoff button and you're ready to ship packages!

Come see me if you have any problems.

-Paul McLaughlin

From: William Smith [IMCEAEX-_O=VELLEB+2C+20INC+2E_OU=VELLEB_CN=RECIPIENTS_CN=WILLIAM@iship.com]
Sent: Saturday, September 19, 1998 1:35 PM
To: Shaindell Goldhaber; David Bennett; Chad Mentzer
Cc: John Dietz
Subject: RE: Proposed menus

An implementation for voiding a package if you are not logged in:
An edit field where you type the iShip tracking number
press a button and it retrieves the least amount of info. for the
user to verify that the package is the one they shipped. At this
point you can void it or reprint the label.

Problems I see:

People messing with are system and trying to guess
tracking numbers and voiding packages or reprinting
another person's labels.

The other implementation is to have the person send us
an email if they want to void a package or reprint a label.
We send them an email that gives them a special number
they enter that only works with that package. We can
check the sender's e-mail against the e-mail address of
the person who shipped the package.

W.

> -----Original Message-----
> From: Shaindell Goldhaber
> Sent: Friday, September 18, 1998 5:26 PM
> To: William Smith; David Bennett; Chad Mentzer
> Subject: Proposed menus
>
> David doesn't like the idea of including left-hand menu buttons for section start pages. However, I've included them here for the sake of discussion. Also, I've added Reprint Label and Void a Package to the shipping menus. I'm not sure what the implementation would look like for users who are not logged on, however.
>
> --Shaindell
>
>
> Logged Off
>
> Home
> Log on
> Welcome
> Sign up
> Help
>
> Shipping
> Log on
> Ship a Package
> Reprint Label (?)
> Void Package (?)
> Compare Services
> Drop off Locator
> Help
>
> Tracking
> Log on

```
>                    Track a Package
>                    File a Claim
>                    Help
>
>          Support
>                    Log On
>                    Contact Us
>                    Feedback
>                    How To
>                    Help
>
> Logged On
>          Home
>                    Log off
>          Welcome
>                    Address Book
>                    Preferences> ...>
>                    Help
>
>          Shipping
>                    Log off
>                    Ship a Package
>                    Reprint a Label
>                    Void a Package
>                    Compare Services
>                    Drop off Locator
>                    Help
>
>          Tracking
>                    Log off
>                    Track a Package
>                    View Shipping Log
>                    File a Claim
>                    Help
>
>          Support
>                    Log Off
>                    Contact Us
>                    Feedback
>                    How To
>                    Help
>
>
```

From: William Smith [IMCEAEX-_O=VELLEB+2C+20INC+2E_OU=VELLEB_CN=RECIPIENTS_CN=WILLIAM@iship.com]
Sent: Wednesday, September 09, 1998 5:44 PM
To: Chad Mentzer
Cc: The Teamsters; The Raiders; The Wolverines
Subject: Configuration Interface Design.

This design can be reused for all objects that need configurations
tied to an OID. Use Gary's config services otherwise.
Paul M - I believe the CONFIGURATION database table is
not being used. It should be removed in favor of either using
Gary's config items or a special configuration tabel modeled
after what Chad is doing.

CHAD:

Have something like this returned for a Configure() method on the acct_IAccount and acct_IUser interface.
There would not be a COCLASS exposed. It is modeled after dbs_IConfigServices. Steal the code.
Default values for new accounts at time of registration should reside in Gary's config services.
Don't store config. values that need have database rules enforced
(for example cascade delete). Add columns to your account and accountuser db tables
in these cases.

Add this method to acct_IUser & acct_IAccount

```
        HRESULT Configuration([out, retval] acct_IConfiguration** ppIConfiguration);
```

Define this interface in your IDL

acct_IConfiguration

```
                object,
                uuid(3CAE1318-A17A-11D1-9636-00A02475D4D9),
                dual,
                helpstring("acct_IConfiguration Interface"),
                pointer_default(unique)

        interface acct_IConfiguration : IDispatch

                [
                        propget,
                        id(1),
                        helpstring("property Item")

                ]
                HRESULT Item([in] BSTR Scope, [in] BSTR ItemName, [in, optional] VARIANT GetDefaultPutDescription,
[out, retval] VARIANT *pVal);
                [
                        propput,
                        id(1),
                        helpstring("property Item")

                ]
                HRESULT Item([in] BSTR Scope, [in] BSTR ItemName, [in, optional] VARIANT GetDefaultPutDescription,
[in] VARIANT newVal);

                        id(2),
                        helpstring("method Delete")
```

```
            HRESULT Delete([in] BSTR Scope, [in] BSTR ItemName);

                    id(3),
                    helpstring("method Exist")

            HRESULT Exist([in] BSTR Scope, [in] BSTR ItemName, [out, retval] int *pExists);

                    id(4),
                    helpstring("method Items")
            ]
            HRESULT Items([in] BSTR Scope, [in] util_IMap *ItemMap);
    };
```

The Items() method takes a map filled with keys (itemnames) that you want retrieved. After the call, the key values are filled in. This is for speed of getting multiple items (preferences).

Scope would be for example:
"DOCUMENTS"
Item
"ShippingLabelDevice"
Value
"\\macarthur\hp5msix"
Description (optional)
"Default label printer"

TWO(2) database tables need to be created.

ACCOUNTCONFIG
and
ACCOUNTUSERCONFIG

These should have the same design as the CONFIGITEMS table with the exeption that keys have to be added (unique key is the composite of the asteriks items):

For ACCOUNTCONFIG

ACCOUNTNO	(VARCHAR 6)
SCOPE	(VARCHAR 38)
ITEM	(VARCHAR 255)
VALUE	(VARCHAR 255)
DESCRIPTION	(VARCHAR 255)

For ACCOUNTUSERCONFIG

USERID	(VARCHAR 35)
ACCOUNTNO	(VARCHAR 6)
SCOPE	(VARCHAR 38)
ITEM	(VARCHAR 255)
VALUE	(VARCHAR 255)
DESCRIPTION	(VARCHAR 255)

Cascade delete should be enforced if the account or accountuser is deleted as appropriate. Having configuration items is OPTIONAL.

William Smith
V.P. Engineering
iShip.com, Inc - The Internet Package Shipper

From: Paul McLaughlin [IMCEAEX-_O=VELLEB+2C+20INC+2E_OU=VELLEB_CN=RECIPIENTS_CN=PAULM@iship.com]
Sent: Monday, November 30, 1998 3:20 PM
To: William Smith; Chad Mentzer; Talal Karkutly
Subject: RE: Columns to add to the portal table.

I need to know exactly, given a SiteOID, how to get the associated Carrier Account.

Currently, I go through the following path in many of my stored procedures to resolve a SiteOID to a CarrierAccount:

SiteAndCarrier -> Account -> AccountAndCarrierAcnt -> CarrierAccount

This works, because the SiteOID is only in the Account table once when the AccountTypeName is either "SiteAccount" or "ParentAccount". After the removal of SiteOID from Account I do not see how I can accomplish the Carrier Account resolution. I can't use AccountAndSite, because that table does not resolve to one Account.

It is almost like we need a SiteAndCarrierAccount table?

-LowRider

> -----Original Message-----
> From: William Smith
> Sent: Monday, November 30, 1998 11:49 AM
> To: Paul McLaughlin; Chad Mentzer; Talal Karkutly
> Subject: RE: Columns to add to the portal table.
>
> Yes, I know that was its original intent,
> but w/o it we can't share accounts across sites.
> Comments?
>
> W.
>
> -----Original Message-----
> From: Paul McLaughlin
> Sent: Monday, November 30, 1998 11:47 AM
> To: William Smith; Chad Mentzer; Talal Karkutly
> Subject: RE: Columns to add to the portal table.
>
> Remember, this was supposed to an exclusion table. I don't see how it can fulfill the role.
>
> -LowRider
>
> -----Original Message-----
> From: William Smith
> Sent: Monday, November 30, 1998 11:40 AM
> To: Paul McLaughlin; Chad Mentzer; Talal Karkutly
> Subject: RE: Columns to add to the portal table.
>
> There is a table called ACCOUNTANDSITE that should
> fulfill this role.
>
> W.
>
> -----Original Message-----
> From: Paul McLaughlin
> Sent: Monday, November 30, 1998 11:34 AM
> To: Chad Mentzer; William Smith; Talal Karkutly
> Subject: RE: Columns to add to the portal table.
>

> If SiteOID is removed from ACCOUNT, then there is no way to determine what carrier accounts belong to a site.
>
> Removing SiteOID from ACCOUNT is going to break EOD, the DOL, and probably rating.
>
> -LowRider
>
> -----Original Message-----
> From: Chad Mentzer
> Sent: Monday, November 30, 1998 11:20 AM
> To: William Smith; Talal Karkutly
> Cc: Paul McLaughlin
> Subject: RE: Columns to add to the portal table.
>
> You are correct. Both the SiteOID and Announcements Flag should be removed.
>
> --
> Chad Mentzer
> iShip.com
> The Internet Package Shipper> (tm)>
> Phone: (425) 602-5032
> Fax: (425) 602-5025
> e-mail: chad@iship.com
>
> compare shipping with iShip.com at http://www.business.msn.com
>
>
> -----Original Message-----
> From: William Smith
> Sent: Saturday, November 28, 1998 12:54 PM
> To: Talal Karkutly
> Cc: Paul McLaughlin; Chad Mentzer
> Subject: Columns to add to the portal table.
>
> PARENTNO varchar(6) - The parent account number for all new user or site accounts added through this portal.
> ACCOUNTPREFIX varchar(3) - The prefix to append before all accounts created via this portal
> DOMAIN varchar(35) - The security domain under which all new users should be created and logged in under.
>
> In looking over the ACCOUNT table, it looks like
> SiteOID
> should be removed. An account doesn't have a site. If we want to have
> an account template for new account users that may have a default site,
> then that should be handled using a special accountuser record to act as a
> template. Chad/PaulM any comments?
>
> In the ACCOUNTUSER table, the announcements flag should be removed since it
> now exists in the ACCOUNTUSERCONFIG table. Chad/PaulM any comments?
>
>
>
> William Smith
> V.P. Engineering
> iShip.com, Inc - The Internet Package Shipper
>
>
>

From: iship [IMCEAEX-_O=VELLEB+2C+20INC+2E_OU=VELLEB_CN=RECIPIENTS_CN=ISHIP@iship.com]
Sent: Tuesday, December 01, 1998 6:44 PM
To: William Smith
Subject: Your Shipping Request # M AZHDY6 DYF2C0

At 3:44 PM on Tuesday, December 1, 1998 you shipped a package
with iShip Package Number M AZHDY6 DYF2C0 via Next Day Air Saver.
to Carl Lewis located in SEATTLE, WA.

You must drop off your package before 5:00 PM on
Tuesday, December 1, 1998 in order for your package to arrive
at its destination by Wednesday, December 2, 1998.

Your selected Drop-Off location is:

UPS Drop Box
bellevue, WA 98005

Your package weight is 10.00 lbs.
The billable weight is 10.00 lbs.
The charge for this package is:

Base Service Charge:
$18.75

Total:
$18.75

To check on the Status of the Package, go to:

http://TESTNOC/ShippingNetwork/trk.asp?T=MAZHDY6DYF2C0

To view any confidential information about the Request/Package you must,
however, go to http://TESTNOC and Log On to your account.

Thank you for shipping with iShip.com!

From:	Paul McLaughlin [IMCEAEX-_O=VELLEB+2C+20INC+ 2E_OU=VELLEB_CN=RECIPIENTS_CN=PAULM@iship.com]
Sent:	Tuesday, July 20, 1999 5:20 PM
To:	iShip Devel
Subject:	Some very interesting ActiveX/IIS issues have been solved

Ordinarily, this might have been a day or two road block, but since Mark, Reichie and I pulled together for about a half an hour, {voice:select("Hanz and Franz")} "We crushed those girly man bugs like the puny insects they are. Ya!" {voice:select ("default")} iShip.com rocks!

Problem 1: ActiveX controls refused to download even though client system was clean.
Solution: The virtual directory "bin" (that points to your equivalent of $/WebApps/bin), had in IIS the "Execute" permission set for the directory, when it should have been only "Script".

Problem 2: CSZ Database ActiveX control installed correctly, but CMS still thought it had not.
Solution: As far as COM was concerned, it did install correctly. All the files where registered and on the client's system. However, the Z5PLUS.TXT needs to be in the same directory as the MSI_ShippingStation.dll and my JScript calls a COM function in the dll to check this. It was this call that was failing for some reason.

So why then was the ActiveX control installing correctly for COM, but the Z5PLUS.TXT file was not in the "downloaded program files" directory when viewed from a command prompt? Well, turns out that when the ActiveX Component Download process checks to see if a file already exists, it doesn't just check the target directory of the file (specified in the .INF file), but rather it does the standard "search for file" process as defined by the Win32 API function SearchPath(). This algorithm searches several different locations, the last of which is the PATH, and guess what file was coincidentally found along the PATH? Yep, Z5PLUS.TXT was in the "...\debug\bin" directory which was in the path (as it is for most of us developers). So, we nuked the Z5PLUS.TXT file from "debug\bin", tried the CMS install again and everything worked fine.

I am going to check to see if my use of the default destination directory causes the search to happen, or if I tell it explicitly to use the "downloaded program files" directory that it will not search for the file in the aforementioned manner. I'll let you know.

Learn something new every day.

Paul R. McLaughlin
iShip.com
Your Internet Package Shipper (tm)
Senior Software Developer
(425)602-5046 (work)
(425)602-5025 (fax)
(425)556-9257 (home)
(425)444-9257 (cell)
e-mail: paulm@iship.com
http://www.iship.com

compare shipping with iShip.com at http://www.business.msn.com/shipping/compare.asp

From: Paul McLaughlin [IMCEAEX-_O=VELLEB+2C+20INC+2E_OU=VELLEB_CN=RECIPIENTS_CN=PAULM@iship.com]
Sent: Wednesday, July 21, 1999 2:57 PM
To: William Smith; iShip Devel
Cc: Ken Sinclair
Subject: RE: Caching Data On The CMS Client:

Importance: High

Here is how it will be done. I will use the SiteInformation dialog as an example throughout:

Overview:
1. The developer in charge of data that needs to be cached on the client, will develop a global JScript object exposing the data as properties and will be contained in the cmsApps.asp client-side script.
2. UI that presents this data, for example the Site Information dialog, will continue to use the tx file they have to get the data, however, the tx file will update the global object (via the object's exposed Update() method) and then the dialog will instead update its UI with the property values from the object.
3. Client-side script that cares about such data will always get it from the global objects.
4. I will write a server and client-side "command processor" that continuously polls our servers for new commands to run, one of which can be "submit a tx file". This will be the exact same tx file that the dialogs use to get the latest data. The client-side command interpreter will then also call the global object's Update() method exactly as in item (2).

Note that the development of the global objects and their integration with the dialogs are not effected by my development of the command interpreters. The dialogs and global objects should be completely unaware of any existence of my command interpreters.

TODO: Detailed Development Tasks
Stuff you need to do:
1a. If you have data that needs to be cached, then you need to create a new ".js" file that contains an object named after your UI component (like William recommends below). The .js file should also be named after the object name.
1b. The JScript object will expose a property for each data element that is cached. For example, gSiteInformation.City and gSiteInformation.State would return the respective data items.
1c. The JScript object will support a member function called Update() which takes the exact same object that is returned by the tx file when it calls its equivalent of "parent.txComplete(txResult)".
1d. All JScript functions in the .JS file will be prefixed with a unique prefix that is named after the object. This is necessary because many .js files will be included in a single page and we must avoid name conflicts. For example, the JScript function for the object's Update function property might be prototyped as "function si_Update(txResult)" and then of course later on it would be assigned to the Update property in the constructor of the SiteInformation object as "this.Update = si_Update;".

2a. cmsApps.asp will be the container for all the cached data.
2b. cmsApps.asp will be modified with "<script language="jscript" src="yourobject.js"></script>" for each object that is needed.

3a. Your UI components current have some sort of "txGet....asp" to retrieve the data. This file will stay the same with one exception: you must rename the "parent.txComplete(txResult);" line of script to be something unique named after your data set. For example: "parent.txCompleteSiteInformation(txResult);"
3b. Move and modify the code of "txCompleteSiteInformation(txResult)" that updates the UI from the txResult object, to a separate function. In that function, instead of getting your data from the txResult object, you must get the data from the properties of the global cached object.
3b. Modify the code of "txCompleteSiteInformation(txResult)" from updating the UI directly, to merely passing the exact same txResult object returned from your tx file, to the cached object's Update() function.

4. All client side script that cares about this cached data, needs to get the data from the appropriate global objects.

Stuff I need to do: (none of which effects your development, but just FYI)
1. Complete command table design and add schema to database.
2. Add data to command tables to facilitate running tx files.
3. Write server-side command processor tx file.

4. Write client-side command interpreter.

NOTE: Until my command interpreter thing works, the object will NOT CONTAIN any data until you open and close the corresponding UI component. In other words, you must open and close the Site Information dialog to "refresh" the data in the SiteInformation global object.

EXAMPLES:

SiteInformation.js:

```
function si_Update(txResult)
{
   // update all data properties with returned tx data.
  this.City = ...["city"];
}

function si_InitProps()

  this.City = "";


function gSiteInformation()

  this.Update = si_Update;
  si_InitProps();


gSiteInformation = new gSiteInformation();
```

cmsApps.asp:

```
...
<script language="JScript" src="/cms/lib/SiteInformation.js"></script>
```

txGetSiteInformation.asp:

```
...
parent.txCompleteSiteInformation(txResult);
```

The SiteInformation dialog:

```
...
function UpdateUI(oSI)
{
  document.frmSI.editCity.value = oSI.City;  // substitute with your objects of course.
  ...


function txCompleteSiteInformation(txResult);
{
  var oSI = dialogArguments.cmsAppsWndPtr.gSiteInformation;  // or however you pass it into your dialog.
  oSI.Update(txResult);
  UpdateUI(oSI);
```

hope this helps. Don't hesitate to ask any questions or send comments my way. They will be appropriately filed. ;-)

-LowRider

> -----Original Message-----
> From: William Smith
> Sent: Tuesday, July 20, 1999 8:58 AM
> To: iShip Devel
> Cc: Ken Sinclair
> Subject: Caching Data On The CMS Client:
>
> This is HIGH priority.
>

> If you worked on the Preferences, Center Information or Scales and Printers dialogs in
> CMS, you must ASAP provide a JavaScript object that exposes the current settings to the
> client side code. Shipping needs access to this information. There is a CMS app area
> that can be used to cache this data globally. Obviously it must remain in sync with changes
> made via the dialogs. One way is to have properties on this object that expose the current values
> and methods that do the transactions. Please send out to the development with sample code,
> how to access the current values on the client side. I would anticipate that a Preferences,
> ScalesPrinters and a SiteInformation object or some other name would exist.
>
> Notice I used SiteInformation NOT CenterInformation. Please think about the fact
> that the code you are creating today will be used in places other than MBE.
>
> Sean Hu has done this very well by creating a JavaScript object for the Hardware settings.
> Sean please work with Reichie and MarkB (if they are the right ones).
>
> PaulM - The COMMAND function you are working on, should have a way to
> refresh these objects on the client side if a change is made on the backend
> using the Administration system.
>
>
> William Smith
> V.P. Engineering
> iShip.com, Inc - Your Internet Package Shipper
>
>
>

From: Jinyue Liu [IMCEAEX-_O=VELLEB+2C+20INC+2E_OU=VELLEB_CN=RECIPIENTS_CN=JINYUE@iship.com]
Sent: Monday, July 26, 1999 2:50 PM
To: Glenn Crowe
Cc: William Smith
Subject: RE: MBE Customer ID Requirement.

The input will be custom id (an integer) and the output will be the check digit.

```
function Mod10(int CustomerID)
{
        int Odd  = 0;
        int Even = 0;
        int i = 0;
        while (CustomerID > 0)
        {
                int Val = CustomerID % 10;

                CustomerID = CustomerID / 10;

                // Separately add up the number in the odd and
                // even positions of the input string.

                if ((i+1)%2)
                        Odd  += Val;
                else
                        Even += Val;

                i = i + 1
        }

        // Add the sum of the odd position characters
        // to twice the sum of the even position characters.
        // Then get the remainder of this sum.

        int CheckDigit = (Odd + (2 * Even)) % 10;

        // The check digit is 0 if the remainder is 0.
        // Otherwise, the check digit is 10 - the remainder.
        if (CheckDigit > 0)
                return '0' + (10 - CheckDigit);
        else
                return '0';
}
```

> -----Original Message-----
> From: Glenn Crowe
> Sent: Friday, July 23, 1999 3:02 PM
> To: Jinyue Liu
> Subject: FW: MBE Customer ID Requirement.
>
> Jinyue,
>
> This is the algorithm that William wants me to implement.
>

> Thanks!
> --
> Glenn Crowe
> Associate Developer
> iShip.com
> Your Internet Package Shipper (tm)
> glennc@iShip.com
> 425.602.5044
>
> Check us out at http://iShip.com.
>
>
> -----Original Message-----
> From: William Smith
> Sent: Friday, July 23, 1999 2:56 PM
> To: David Bennett
> Cc: Glenn Crowe
> Subject: MBE Customer ID Requirement.
>
> The customer id will be a 15 character field with the following format:
>
> MCMS###########
>
> The human readable format shall be:
>
> MCMS-###-####-####-#
>
> The prefix "MCMS" stands for MBE Counter Manifest System
>
> The middle 10 digits are 0 padded integer starting at 1.
> Each newly generated Customer # will be a simple increment
> to the last number assigned customer number.
> The human readable portion of this number is hyphenated just
> like a phone number that includes the area code.
>
> The final digit is a MOD 10 (UPS Algorithm) of the 10 digit number
> portion of the customer id. The Check digit does not include the
> "MCMS" prefix.
>
> In all cases where the customer id to read in the CMS, EPSO applications or
> on a report, the human readable version shall be display.
>
> In all cases where a customer id can be entered into the system, the data
> entry field shall support both the human readable version and the raw
> customer id. Data entry shall except either hyphens or spaces the
> delimit fields.
>
> Implementation notes:
> The current value of the MBE customer id will be stored in the _AccountConfig database
> table associated with the MBE master iShip Account#.
>
> The stored procedure that retrieves the next available customer id shall take an iShip Account# as
> input. This account# shall be used to locate the AccountConfig item. If this item does not exist,
> it shall be created, and a customer id with the number portion set to "1" shall be returned.
>
> The stored procedure shall return a complete customer id in non-human readable format as a string.
>
> The stored procedure must make sure the deadlocks cannot occur and the transaction is atomic.
>
> The account config item shall be named:
> MBE.CUSTOMERID.CURRENT
>
> William Smith
> V.P. Engineering

> iShip.com, Inc - Your Internet Package Shipper
>
>
>